Week2AA

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.2 ✓ purrr 0.3.4  
## ✓ tibble 3.0.4 ✓ dplyr 1.0.2  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.4.0 ✓ forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

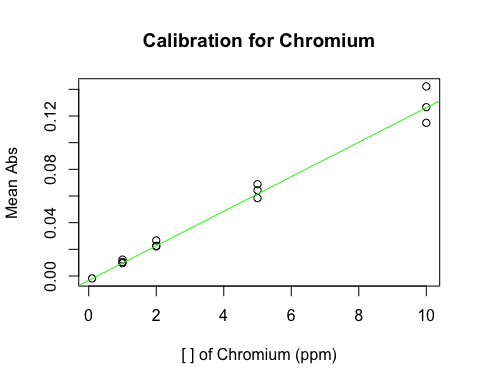
library(readr)

AA<-read\_csv("~/ICPMSArielle/Data/AA\_DATA.csv") %>%  
 mutate(percent\_rsd = as.numeric(percent\_rsd)\*100)

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## X1 = col\_double(),  
## sample\_key = col\_double(),  
## mean\_abs = col\_double(),  
## percent\_rsd = col\_character(),  
## type = col\_character(),  
## site = col\_character(),  
## analyst = col\_character(),  
## mass\_of\_soil = col\_double(),  
## total\_volume = col\_double(),  
## concentration = col\_double()  
## )

sample\_sites <- unique(filter(AA, site != "MB", site != "")$site)

#filtering for a single metal then selecting variables of interest  
cal <- AA %>%  
 filter(type != "Sample", percent\_rsd != "HIGH") %>%  
 select(mean\_abs, percent\_rsd, concentration)  
#weighted linear regression  
w <- 1/(cal$mean\_abs\*cal$percent\_rsd)^2  
model <- lm(cal$mean\_abs ~ cal$concentration, weights = w)  
slope <- model$coefficients[2]  
intercept <- model$coefficients[1]  
slope\_std <- summary(model)$coefficients[2,2]  
intercept\_std <- summary(model)$coefficients[1,2]  
plot(cal$mean\_abs ~ cal$concentration,  
 xlab = paste("[ ] of Chromium (ppm)"),  
 ylab = "Mean Abs") +  
 abline(model, col = "green") +  
 title(paste("Calibration for Chromium"))



## integer(0)

equation <- tibble(metal = "Chromium", slope, slope\_std, intercept, intercept\_std)  
cal <- rbind(equation)  
cal

## # A tibble: 1 x 5  
## metal slope slope\_std intercept intercept\_std  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Chromium 0.0130 0.000314 -0.00328 0.000621

##Same as ICPMS from James code, adjusted for AA dataset  
sample\_analysis <- function(unique\_site){   
#initiated "for loops" to select a specific sample and a specific metal  
 concentration\_data <- NULL  
 sample <- filter(AA, site == unique\_site)  
 data <- NULL  
 for (ID in sample$sample\_key){  
 sample\_dataAA <- filter(sample, sample\_key == ID)  
 #convert the cps readings into concentrations  
 #sample analysis  
 m <- cal$slope  
 b <- cal$intercept  
 y <- sample\_dataAA$mean\_abs  
 b\_e <- cal$intercept\_std  
 m\_e <- cal$slope\_std  
 x <- (y-b)/m #The units are dependent on the calibration standards (Kg/mL)  
   
 RSD <- ((sample\_dataAA$percent\_rsd/100)\*sample\_dataAA$mean\_abs)  
 absorption <- sample\_dataAA$mean\_abs  
 #propagate the error in the concetration  
 #error propagation  
 e\_yb <- sqrt((RSD)^2 + (b\_e)^2) #error in y-b from calibration  
 yb <- absorption - b  
 e\_x <- x\*sqrt((e\_yb/yb)^2 +(m\_e/m)^2)   
 #error in x from calibration  
 #if the "site is not eh method blank, store the concentration data  
 data <- rbind(data, data\_frame(sample\_key = ID, x, e\_x))  
 if(unique\_site != "MB"){  
 concentration\_data <- data\_frame(sample\_key = sample\_dataAA$sample\_key,  
 analyst = sample\_dataAA$analyst,  
 metal = "Chromium",  
 site = unique\_site,  
 conc\_dil = x,  
 conc\_dil\_error = e\_x) %>%  
 rbind(concentration\_data)  
 }  
 #if the site is the method blank, average the concentrations and then store the data  
 if (unique\_site == "MB"){  
 x <- mean(data$x)  
 e\_x <- sd(data$x)  
 concentration\_data <- data\_frame(metal = "Chromium",  
 site = unique\_site,  
 conc\_dil = x,  
 conc\_dil\_error = e\_x) %>%  
 rbind(concentration\_data)  
 }  
 }  
   
   
 return(concentration\_data)  
}

#create a function that runs a different function on each of the soil sample sites  
  
#inputs: a function  
#outputs: a data frame with the function outputs from each site  
  
run\_sites <- function(Function){  
 value <- NULL  
 for (site in sample\_sites){  
 site\_value <- Function(site)  
 value <- rbind(site\_value, value)  
 }  
 return(value)  
}

#analyze the method blank and all the samples  
MB<- sample\_analysis("MB") #ug/kg

## Warning: `data\_frame()` is deprecated as of tibble 1.1.0.  
## Please use `tibble()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

uncor\_sample<- run\_sites(sample\_analysis)  
#values do not account for dilutns  
MB

## # A tibble: 10 x 4  
## metal site conc\_dil conc\_dil\_error  
## <chr> <chr> <dbl> <dbl>  
## 1 Chromium MB 0.142 0.0462  
## 2 Chromium MB 0.146 0.0472  
## 3 Chromium MB 0.148 0.0500  
## 4 Chromium MB 0.144 0.0526  
## 5 Chromium MB 0.144 0.0576  
## 6 Chromium MB 0.140 0.0638  
## 7 Chromium MB 0.126 0.0632  
## 8 Chromium MB 0.0961 0.0271  
## 9 Chromium MB 0.110 0.0164  
## 10 Chromium MB 0.122 NA

uncor\_sample

## # A tibble: 34 x 6  
## sample\_key analyst metal site conc\_dil conc\_dil\_error  
## <dbl> <chr> <chr> <chr> <dbl> <dbl>  
## 1 56 AVM Chromium A 0.778 2.81   
## 2 54 LAK Chromium A 0.739 1.61   
## 3 48 AH Chromium A 1.27 4.71   
## 4 46 LML Chromium A 1.04 3.20   
## 5 58 SS Chromium B 0.863 1.71   
## 6 52 MF Chromium B 0.716 1.92   
## 7 45 KAD Chromium B 0.747 1.18   
## 8 41 LF Chromium B 0.701 0.736  
## 9 40 AB Chromium B 0.631 1.68   
## 10 6 MRMJ Chromium C 0.917 4.83   
## # … with 24 more rows

sample\_data\_mb <- NULL  
  
 conc\_dil\_blanked<- uncor\_sample$conc\_dil-MB$conc\_dil

## Warning in uncor\_sample$conc\_dil - MB$conc\_dil: longer object length is not a  
## multiple of shorter object length

#error propogation: subtraticoon of MB  
 conc\_dil\_blanked\_error <- sqrt((uncor\_sample$conc\_dil\_error)^2 +(MB$conc\_dil\_error)^2)

## Warning in (uncor\_sample$conc\_dil\_error)^2 + (MB$conc\_dil\_error)^2: longer  
## object length is not a multiple of shorter object length

sample\_data\_mb<-uncor\_sample%>%  
 mutate(conc\_dil\_blanked, conc\_dil\_blanked\_error)%>%  
 rbind(sample\_data\_mb)  
  
  
sample\_data\_mb

## # A tibble: 34 x 8  
## sample\_key analyst metal site conc\_dil conc\_dil\_error conc\_dil\_blanked  
## <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 56 AVM Chro… A 0.778 2.81 0.636  
## 2 54 LAK Chro… A 0.739 1.61 0.594  
## 3 48 AH Chro… A 1.27 4.71 1.12   
## 4 46 LML Chro… A 1.04 3.20 0.897  
## 5 58 SS Chro… B 0.863 1.71 0.719  
## 6 52 MF Chro… B 0.716 1.92 0.576  
## 7 45 KAD Chro… B 0.747 1.18 0.621  
## 8 41 LF Chro… B 0.701 0.736 0.605  
## 9 40 AB Chro… B 0.631 1.68 0.521  
## 10 6 MRMJ Chro… C 0.917 4.83 0.795  
## # … with 24 more rows, and 1 more variable: conc\_dil\_blanked\_error <dbl>

#define the dilution factors and measuremnt errors  
#error propagation  
vol\_e <- 1  
mass\_e <- 0.001  
dil\_1010\_e <- sqrt(1^2 + 10^2)  
dil\_e <- sqrt((dil\_1010\_e/1010)^2 + (1/10)^2) #error in 101 dilution factor  
  
#correct for dilutions and propagate error  
sample\_data <- merge(AA, sample\_data\_mb) %>%  
 unique() %>%  
 mutate(conc\_blanked = conc\_dil\_blanked\*(total\_volume/1000)/(mass\_of\_soil/1000),  
 #101 is the factor diluted by at OHSU to make the solutions dilute enought to run the ICPMS on  
 conc\_blanked\_error = conc\_blanked \*   
 sqrt((conc\_dil\_blanked\_error/conc\_dil\_blanked)^2 +   
 (dil\_e/101)^2 +  
 (mass\_e/mass\_of\_soil)^2 +  
 (vol\_e/total\_volume)^2),  
 conc\_unblanked = conc\_dil\*(total\_volume/1000)/(mass\_of\_soil/1000)\*101,  
 conc\_unblanked\_error = conc\_unblanked\*  
 sqrt((conc\_dil\_error/conc\_dil)^2 +  
 (dil\_e/101)^2 +  
 (mass\_e/mass\_of\_soil)^2 +  
 (vol\_e/total\_volume)^2)) %>%  
 select(-concentration, #removing uneccesary columns  
 -type,  
 -mass\_of\_soil,  
 -total\_volume,  
 -mean\_abs,  
 -percent\_rsd,  
 -conc\_dil\_blanked,  
 -conc\_dil\_blanked\_error,  
 -conc\_dil,  
 -conc\_dil\_error)

rm(list=ls()[!(ls()%in% c("AA", "sample\_data"))])

AVG\_concSITE<- sample\_data%>%  
 filter(site!= "QC")%>%  
 group\_by(site)%>%  
 summarise(mean\_conc = mean(conc\_blanked), sd\_conc = sd(conc\_blanked), n = n()) %>%  
 mutate(CI = qnorm(0.975)\*sd\_conc/sqrt(n),  
 lower\_ci = mean\_conc - CI,  
 upper\_ci = mean\_conc + CI)

## `summarise()` ungrouping output (override with `.groups` argument)

AVG\_concQC<- sample\_data%>%  
 filter(site== "QC")%>%  
 group\_by(site)%>%  
 summarise(mean\_conc = mean(conc\_blanked), sd\_conc = sd(conc\_blanked), n = n()) %>%  
 mutate(CI = qnorm(0.975)\*sd\_conc/sqrt(n),  
 lower\_ci = mean\_conc - CI,  
 upper\_ci = mean\_conc + CI)

## `summarise()` ungrouping output (override with `.groups` argument)

AVG\_concALL<- sample\_data%>%  
 filter(site!= "QC")%>%  
  
 summarise(mean\_conc = mean(conc\_blanked), sd\_conc = sd(conc\_blanked), n = n()) %>%  
 mutate(CI = qnorm(0.975)\*sd\_conc/sqrt(n),  
 lower\_ci = mean\_conc - CI,  
 upper\_ci = mean\_conc + CI)  
  
write.csv(sample\_data, "~/ICPMSArielle/Data/AA\_data2.csv")  
  
write.csv(AVG\_concSITE, "~/ICPMSArielle/Data/AAAVG\_concSITE.csv")

t.test2 <- function(m1,m2,s1,s2,n1,n2,m0=0,equal.variance=FALSE)  
{  
 if( equal.variance==FALSE )   
 {  
 se <- sqrt( (s1^2/n1) + (s2^2/n2) )  
 # welch-satterthwaite df  
 df <- ( (s1^2/n1 + s2^2/n2)^2 )/( (s1^2/n1)^2/(n1-1) + (s2^2/n2)^2/(n2-1) )  
 } else  
 {  
 # pooled standard deviation, scaled by the sample sizes  
 se <- sqrt( (1/n1 + 1/n2) \* ((n1-1)\*s1^2 + (n2-1)\*s2^2)/(n1+n2-2) )   
 df <- n1+n2-2  
 }   
 t <- (m1-m2-m0)/se   
 dat <- c(m1-m2, se, t, 2\*pt(-abs(t),df))   
 names(dat) <- c("Difference of means", "Std Error", "t", "p-value")  
 return(dat)   
}

t.test2(48.91812,23.05865445, 9.882947,7.41209171, 12, 4 )

## Difference of means Std Error t p-value   
## 2.585947e+01 4.676982e+00 5.529092e+00 9.034056e-04

t.test2(48.91812,16.151985, 9.882947,5.164864565, 12, 5 )

## Difference of means Std Error t p-value   
## 3.276614e+01 3.670770e+00 8.926230e+00 4.138460e-07

t.test2(48.91812,26.61849284, 9.882947,13.26031917, 12, 5 )

## Difference of means Std Error t p-value   
## 22.29962716 6.58077501 3.38860197 0.01489155

t.test2(48.91812,9.840144961, 9.882947,4.102673005, 12, 3 )

## Difference of means Std Error t p-value   
## 3.907798e+01 3.708103e+00 1.053854e+01 3.048136e-06

t.test2(48.91812,13.34216869, 9.882947,7.130288564, 12, 2 )

## Difference of means Std Error t p-value   
## 35.57595131 5.79309022 6.14110086 0.03585291

t.test2(48.91812,6.94623958, 9.882947,5.13474049, 12, 3 )

## Difference of means Std Error t p-value   
## 4.197188e+01 4.114354e+00 1.020133e+01 3.348115e-05

t.test2(48.91812,17.41502, 9.882947,10.24718, 12, 22 )

## Difference of means Std Error t p-value   
## 3.150310e+01 3.593373e+00 8.767000e+00 7.292574e-09

#f-test for the icpms vs the aa  
t.test2(48.91812,17.41502, 9.882947, 10.24718, 12,22 )

## Difference of means Std Error t p-value   
## 3.150310e+01 3.593373e+00 8.767000e+00 7.292574e-09